ディープラーニングへの Ruby 適用試行に関する報告

2017 年 2 月 15 日

Japan OSS Promotion Forum
アプリケーション部会
サイオステクノロジー　株式会社
手塚　拓

目次

1. ディープラーニングとは
2. ディープラーニングに Ruby を利用する価値
3. 「Ruby でディープラーニング」の問題点
4. 現状報告
   I. 予備知識
   II. 検証
   III. 報告
5. 劣勢は跳ね返せるのか？

1. ディープラーニングとは

- **機械学習の一種**
  - 多層構造のニューラルネットワーク（ディープニューラルネットワーク、英: deep neural network）を用いた機械学習
  - ニューラルネットワークの多層化、特に3層以上のものに対し、1990年代に進められた脳、特に視覚野の研究や、「たった一つの学習理論（英語: One Learning Theory）」、ブルーノ・オルスホーゼンによるスパース・コーディング理論を基にしたアルゴリズムが実装されたものを指す
  - AlphaGo(*1) 等の登場により、近年高注目度の IT 技術

*1: Google DeepMindによって開発されたコンピュータ囲碁プログラム
2015 年 10月に、人間のプロ囲碁棋士を互先（ハンディキャップなし）で破った初のコンピュータ囲碁プログラム

2. ディープラーニングに Ruby を利用する価値

- **言語機能として、ディープラーニングシステムを汲み上げられ得るか？**

↓

Yes(*2)

*2: すでに、RubyBrain（ https://github.com/elgoog/ruby_brain ）と言う名の、
Ruby 製 ディープラーニングフレームワークが作成されている等の実績がある

- **他のプログラミング言語に比べ、ディープラーニングシステムを容易に組み上げることが可能か？**

↓

No

（ある Ruby コミッタ曰く「Python と比べて 11 年の遅れがある」）

3.「Ruby でディープラーニング」の問題点

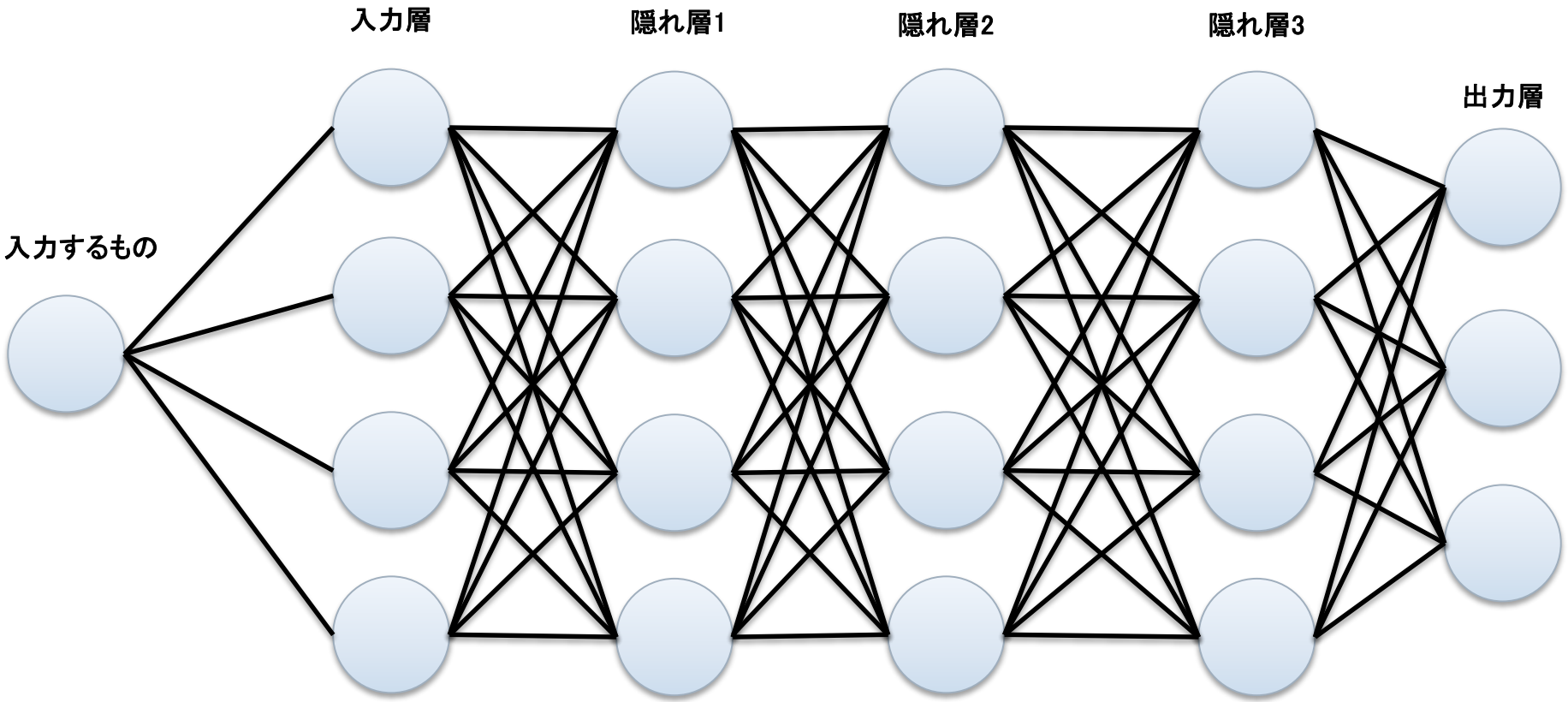
Japan OSS Promotion Forum

- **ハード対応的問題点**
  - GPU等、CPU以外の計算リソースの有効利用に難がある
  - 多コア CPU の有効利用に難がある

- **ライブラリ・ツール観点での問題点**
  - ディープラーニングシステムを容易にくみ上げるためには、関連ライブラリ（特に行列変換等高度数値計算ライブラリ）やツールの有無が鍵
  - Python は、NumPy と言うデファクトな数値計算ライブラリを中核に、ライブラリやツールの整備が行われている
  - Ruby はデファクトな数値計算ライブラリが存在せず、各ライブラリ・ツールごとに使用している数値計算ライブラリが異なり、結果ツール同士のデータ連携等に難がある

4.現状報告:予備知識:深層学習とは

- 脳機能における特性をコンピュータ上でシュミレーションできることを目指した多層構造の機械学習手法の一種。
- 機械自身が特徴抽出できる技術の事をいう。



4.現状報告:予備知識:活性化関数

- 活性化関数とはニューロンの出力値を決める関数の事です。 Sigmoid 関数の他に tanh 関数や linear 関数などが存在します。
- Sigmoid関数では負の値を取ることが出来ず、 0 ～ 1 の値を学習データとして渡す必要があるが、活性化関数の中には負の値を取ることが出来るものも存在するため、他の活性化関数を利用する場合は -1 ～ 1 の値で学習データを渡す事が出来きます
- この当たりの話は実は深層学習の話をする上でも重要な事だと思うのですが、各種関数アルゴリズムについて説明し始めると時間が足りなくなると思いますので、今回の検証では触れないものとします。

4.現状報告:予備知識:CNN、RNN

- **CNNは折り畳みニューラルネットワークと呼ばれ画像を分類する場合などに利用する際に用います。**
  - **例えば今回の検証2で利用するMNISTはCNNで実装しクラス分類します。**

- **RNNは再帰ニューラルネットワークと呼ばれ、数値を出力する時や入力予測で用います。**
  - **例えば今回の検証1で利用するSIN波形の近似予測はRNNで実装し予測します。**

4.現状報告:検証用ライブラリについて

- **Ruby : RubyBrain を選択**
- **Python : Keras を選択**

- **選択理由（RubyBrain）**
  - **Qiitaに使い方が日本語で記載されていた事が大きい**
  - **アプリケーション部会内で情報として共有されたのもRubyBrainだった**
  - **検証者がRubyに明るい方ではないので扱いやすそうだという直感もあり。**
  - **他の選択肢としては　AI4R などが上がると思われます**

- **選択理由(Keras)**
  - **検証開始以前に使っていたライブラリである事が大きい**
  - **他の選択肢としては最近の話題的に Tensorflow や chainer あたりだと思われる**
  - **Keras は Tensorflow とTheano をラップする形で実装されているらしく、計算部分についてはTensorflow, Theanoから選択可能**

4.現状報告:RubyBrainについて

- **Ruby用のdeep learningライブラリ**
- **Rubyの組み込み＆標準ライブラリのみ使用して実装している**
- **ニューラルネットワークを行う場合、活性化関数としてsigmoid関数を持つニューロンしか実装されていないため、負の値を取ることが出来ないらしい。**
- **実装されている手法が明確ではないが、全結合ニューラルネットワークだと思われる**
- **2016年10月20日現在の最新バージョンv0.1.4**
- **詳しくは参考URLを参照**

4.現状報告：RubyBrain-sin波形近似予測：準備

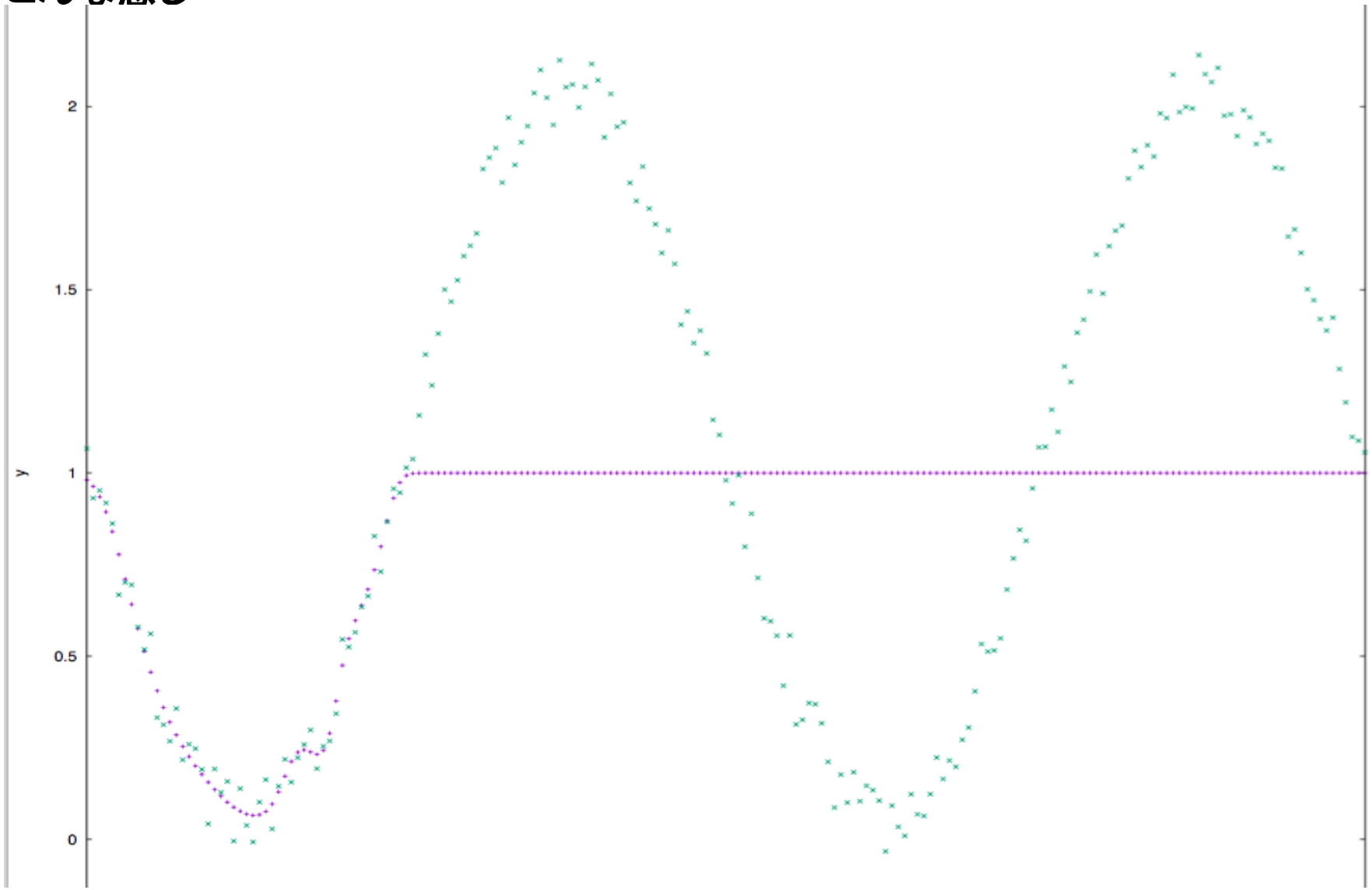
- **Ruby2.3.0**
- **RubyBrainを利用**
- **独自に作成したsin波形の予測を行う**
  - **データセットsin波形は360°分を利用する**
- **前提条件**
  - **Intel Core i7 Mac2012年モデルを利用する**
  - **Sin波形は qiita.com/elgoog/items/8e7102a87889950d060dを参考にする**
  - **ニューラルネットワークの層は4層とする**
- **ニューラルネットワークの隠れ層は2層とする。その際の隠れ層1は24ニューロン、隠れ層2は12ニューロンとした。**
- **Epoch（繰り返し回数）は5000回にセットし、離脱するレートを0.005とした。**

4.現状報告:Keras-sin波形近似予測:準備

- 比較としてPythonのKerasライブラリを利用した場合を記する
- データについてはRubyBrainで利用したものとほぼ同等のものを実装してデータ生成する
- 前提条件
  - Intel Core i7 Mac 2012年モデル
  - ニューラルネットワーク層は3層として、隠れ層（中間層）は300ニューロン準備する
  - 本来このような予測行う場合はLSTMと呼ばれるRNNを選択するが、今回はRubyBrainにLSTMが実装されていないので選択せず。
- Epoch（繰り返し回数）は40000回にセットし、validation_split(テストデータとする実データの割合)を5%とした
- EarlyStoppingを実装しているため、学習結果に一定のルールが存在した段階で学習をやめるようになっている

4.現状報告:RubyBrain-sin波形近似予測:実測

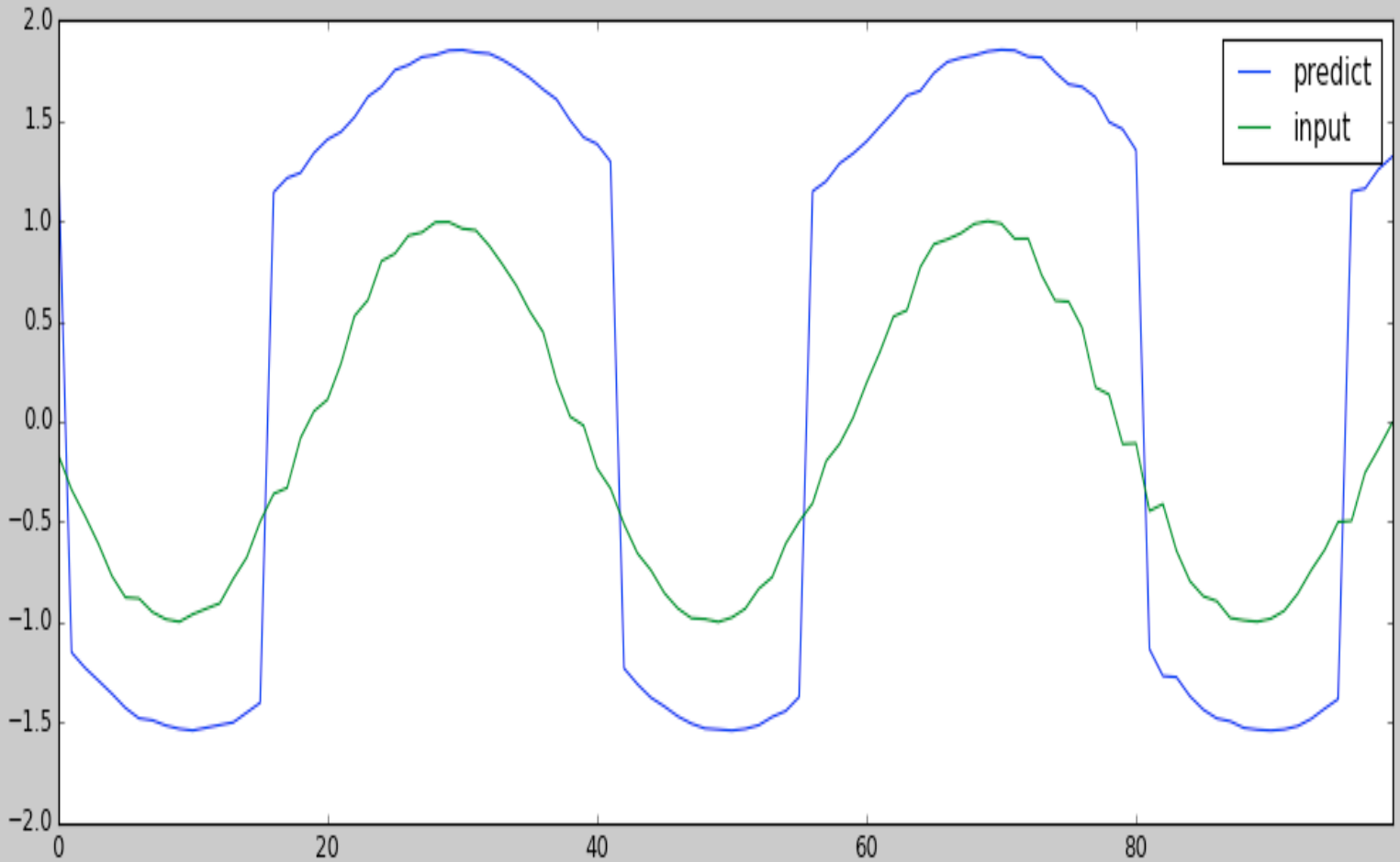
- こんな感じ



4.現状報告:Keras-sin波形近似予測:実測

Japan OSS Promotion Forum

- こんな感じ



4.現状報告：RubyBrain-MNIST:準備

- Ruby2.3.0
- RubyBrainを利用
- RubyBrainから取得可能なMNISTデータを利用する
- 前提条件
  - Intel Core i7 Mac2012年モデルを利用する
  - MNIST実装は quiita.com/elgoog/items/8e7102a87889950d060dを参考にする
- 入力層のニューロンは28x28＝784ニューロンとします
- ニューラルネットワークの隠れ層は1層とする。その際の隠れ層1は128ニューロンとします
- 出力層のニューロンは0~9のラベルを付与するため10ニューロンとします
- Epoch（繰り返し回数）は15回にセットし、離脱するレートを0.005とした。

4.現状報告:Keras-MNIST:準備

- データについてはSciPyがデータセットとして持っているMNISTデータを利用する
- 前提条件
  - Intel Core i7 Mac 2012年モデル
  - ニューラルネットワーク層は3層として、隠れ層（中間層）は128ニューロン準備する。
- 入力層のニューロンは28x28＝784ニューロンとします
- ニューラルネットワークの隠れ層は1層とする。その際の隠れ層1は128ニューロンとします
- 出力層のニューロンは0~9のラベルを付与するため10ニューロンとします
- Epoch（繰り返し回数）は30回にセットし、validation_split(テストデータとする実データの割合)を5%とした。
- EarlyStoppingを実装しているため、学習結果に一定のルールが存在した段階で学習をやめるようになっている

4.現状報告:計測概要

項目	RubyBrain	Keras
波形近似-精度	▲(180°分しか予測できないが180°分については正答率が高いと思われる)	▲(360°分予測出来、且つ正解率が50%～60%程度)
波形近似-平均離脱学習回数	(処理時間の関係で1回実施)	(5回実施)
波形近似-速度	(別表参照)	(別表参照)
MNIST-精度	N/A	○
MNIST-速度	N/A	(別表参照)
MNIST-平均離脱学習回数	N/A	○

4.現状報告:SIN波形近似予測

項目	RubyBrain	Keras
波形近似-精度	だいたい50%	だいたい50~75%
波形近似-速度	32737.8305 secs.	2283.2203 secs. 1024.8873 secs. 705.1122 secs. 1035.3644 secs. 1503.0669 secs.
波形近似-平均離脱学習回数	Epoch 39999/40000	Epoch 666/40000 Epoch 114/40000 Epoch 201/40000 Epoch 292/40000 Epoch 194/40000

4.現状報告:MNIST

項目	RubyBrain	Keras
MNIST-精度	計測不可	5回分の試行で平均約90%
MNIST-速度	計測不可 ※2 業務中に完了しない	5回分の試行で541～551secの間で完了
MNIST-平均離脱学習回数	計測不可（30/30回らず）	30/30 30/30 30/30 30/30 30/30

4.現状報告:比較コメント(私観)

項目	RubyBrain(Ruby)	Keras(Python)
導入のしやすさ	gem install ruby_brainで可能 シンプルで簡単	Numpy, Theano or ThensorFlowなどの影響パッケージを入れないと導入不可
ドキュメントの多さ	少ないと思われる	比較的整っている
予測精度	学習方式が1種類のため時系列に弱いと思われるがそれなりの精度があると思われる	学習方式の種類によって違うがマッチしたものを使うと80～90%程度であれば出すことも可能(サンプルなら)
速度(後で速度を出しておく)	遅い	RubyBrainに比べると早いと思われる
今回導入したライブラリでの充実度(利用関数の多さなど)	CNN・RNNの各手法が実装されていないように思われる	GRU、LSTM、SimpleRNNなどが実装されている。
複雑度	実装されているものがシンプルなので覚える事が少なく実装しやすいと思われる	比較的覚える事が多く難しい

4.現状報告:比較コメント(私観)

項目	RubyBrain(Ruby)	Keras(Python)
過学習について	過学習を引き起こさないようにするためのメソッドなどは準備されていないため、自分で調整する必要がある	Dropoutというクラスが準備されているので、プログラム内で制御可能
早期停止について	実装されている。val_loss値が設定値以下であった場合にストップさせる事が出来る	EarlyStoppingクラスが準備されているのでプログラムないで制御可能。こちらもval_loss値を元にストップさせていると思われる
作成モデルの保存・復旧	それらしいメソッドが存在する。Yaml形式でweightを持つ形式らしい	保存、復元が可能

5.劣勢は跳ね返せるのか？

- **跳ね返そうとする Ruby コミュニティの存在**
  - SciRuby（科学技術計算 + Ruby に寄与することを目的としたコミュニティ https://sciruby-jp.github.io/ ）等、「Ruby を利用したプログラミングで、デファクトになりうる科学技術計算ライブラリやツールの開発を、日々行っているコミュニティが存在する
- **堅実なプランに基づく活動の存在**
  - Python 等ですでに実現された便利な機能群を、Ruby から利用できるようにする（gem 化等）活動がスタートしている
    - すでに実績のある、便利な機能を Ruby から使えるようにする
    - 全てを一から Ruby で作成し、Python 等と同等の機能を作りこむことは可能だろうが、出来上がるまでに Ruby は見限られる
  - 2017 年度中にある程度劣勢を跳ね返すためのスケジュールも立案されている(*3)

*3:「Development of machine learning infrastructures for Ruby ecosystem」参照

参考.参考URL、書籍

- Rubyでディープラーニング
  - **https://qiita.com/elgoog/items/8e7102a87889950d060d**
- **Keras HP**
  - **https://keras.io/ja/**
- **猫でもわかる機械学習、深層学習入門と重要性**
  - **http://www.ochearno.net/entry/2016/05/15/115028**
- **書籍：ゼロから作るDeepLearning（O'REILLY）**
- **書籍：はじめての深層学習プログラミング**

他多数